Final Project - Analyzing Sales Data

Date: 14 August 2023

Author: Kittipoom Jeannumwong (Bank)

Course: `Pandas Foundation`

```
# import data
import pandas as pd
df = pd.read_csv("sample-store.csv")
```

```
# preview top 5 rows
df.head()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City	...	Postal Code	Region	Product ID	Category	Sub-Category	P N
0	1	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420.0	South	FUR-BO-10001798	Furniture	Bookcases	B S C B
1	2	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420.0	South	FUR-CH-10000454	Furniture	Chairs	H F U S C
2	3	CA-2019-138688	6/12/2019	6/16/2019	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036.0	West	OFF-LA-10000240	Office Supplies	Labels	A A L T b
3	4	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311.0	South	FUR-TA-10000577	Furniture	Tables	B C S R T
4	5	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311.0	South	OFF-ST-10000760	Office Supplies	Storage	E 'M S

5 rows × 21 columns

```
# shape of dataframe
df.shape
```

```
(9994, 21)
```

```
# see data frame information using .info()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Row ID          9994 non-null   int64
 1   Order ID        9994 non-null   object
 2   Order Date      9994 non-null   object
 3   Ship Date       9994 non-null   object
 4   Ship Mode       9994 non-null   object
 5   Customer ID     9994 non-null   object
 6   Customer Name   9994 non-null   object
 7   Segment         9994 non-null   object
 8   Country/Region  9994 non-null   object
 9   City            9994 non-null   object
 10  State           9994 non-null   object
 11  Postal Code     9983 non-null   float64
 12  Region          9994 non-null   object
 13  Product ID      9994 non-null   object
 14  Category        9994 non-null   object
```

We can use `pd.to_datetime()` function to convert columns 'Order Date' and 'Ship Date' to datetime.

```
# example of pd.to_datetime() function
pd.to_datetime(df['Order Date'].head(10), format='%m/%d/%Y')
```

```
0   2019-11-08
1   2019-11-08
2   2019-06-12
3   2018-10-11
4   2018-10-11
5   2017-06-09
6   2017-06-09
7   2017-06-09
8   2017-06-09
9   2017-06-09
Name: Order Date, dtype: datetime64[ns]
```

```
# TODO - convert order date and ship date to datetime in the original dataframe

df['Order Date'] = pd.to_datetime(df['Order Date'], format='%m/%d/%Y')
df['Ship Date'] = pd.to_datetime(df['Ship Date'], format='%m/%d/%Y')

df.head(10)
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City	...	Postal Code	Region	Product ID	Category	Sub-Category
0	1	CA-2019-152156	2019-11-08	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420.0	South	FUR-BO-10001798	Furniture	Bookcases
1	2	CA-2019-152156	2019-11-08	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420.0	South	FUR-CH-10000454	Furniture	Chairs
2	3	CA-2019-138688	2019-06-12	2019-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036.0	West	OFF-LA-10000240	Office Supplies	Labels
3	4	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311.0	South	FUR-TA-10000577	Furniture	Tables
4	5	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311.0	South	OFF-ST-10000760	Office Supplies	Storage
5	6	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	...	90032.0	West	FUR-FU-10001487	Furniture	Furnishing
6	7	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	...	90032.0	West	OFF-AR-10002833	Office Supplies	Art
7	8	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	...	90032.0	West	TEC-PH-10002275	Technology	Phones
8	9	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	...	90032.0	West	OFF-BI-10003910	Office Supplies	Binders
9	10	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	...	90032.0	West	OFF-AP-10002892	Office Supplies	Appliances

10 rows × 21 columns

```
# TODO - count nan in postal code column

df['Postal Code'].isna().sum()
```

```
11
```

```
# TODO - filter rows with missing values

df [ df['Postal Code'].isna() ]
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City	...	Postal Code	Region	Product ID	Category	Sub-Categor
2234	2235	CA-2020-104066	2020-12-05	2020-12-10	Standard Class	QJ-19255	Quincy Jones	Corporate	United States	Burlington	...	NaN	East	TEC-AC-10001013	Technology	Accesso
5274	5275	CA-2018-162887	2018-11-07	2018-11-09	Second Class	SV-20785	Stewart Visinsky	Consumer	United States	Burlington	...	NaN	East	FUR-CH-10000595	Furniture	Chairs
8798	8799	US-2019-150140	2019-04-06	2019-04-10	Standard Class	VM-21685	Valerie Mitchum	Home Office	United States	Burlington	...	NaN	East	TEC-PH-10002555	Technology	Phones
9146	9147	US-2019-165505	2019-01-23	2019-01-27	Standard Class	CB-12535	Claudia Bergmann	Corporate	United States	Burlington	...	NaN	East	TEC-AC-10002926	Technology	Accesso
9147	9148	US-2019-165505	2019-01-23	2019-01-27	Standard Class	CB-12535	Claudia Bergmann	Corporate	United States	Burlington	...	NaN	East	OFF-AR-10003477	Office Supplies	Art
9148	9149	US-2019-165505	2019-01-23	2019-01-27	Standard Class	CB-12535	Claudia Bergmann	Corporate	United States	Burlington	...	NaN	East	OFF-ST-10001526	Office Supplies	Storage
9386	9387	US-2020-127292	2020-01-19	2020-01-23	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Burlington	...	NaN	East	OFF-PA-10001970	Office Supplies	Paper
9388	9389	US-2020-127292	2020-01-19	2020-01-23	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Burlington	...	NaN	East	OFF-AP-10000828	Office Supplies	Applianc
9389	9390	US-2020-127292	2020-01-19	2020-01-23	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Burlington	...	NaN	East	OFF-EN-10001509	Office Supplies	Envelope
9741	9742	CA-2018-117086	2018-11-08	2018-11-12	Standard Class	QJ-19255	Quincy Jones	Corporate	United States	Burlington	...	NaN	East	FUR-BO-10004834	Furniture	Bookcas

11 rows × 21 columns

```
# TODO - Explore this dataset on your owns, ask your own questions

## Note : It's seem had some problem occur after covert order date and ship date to datetime
          ## In some time Show as "NaT" , In some time can passed all
```

Data Analysis Part

Answer 10 below questions to get credit from this course. Write `pandas` code to find answers.

```
# TODO 01 - how many columns, rows in this dataset

## ANS: 21 Columns / 9994 Row

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype         
---  ------          --------------  -----         
 0   Row ID          9994 non-null   int64         
 1   Order ID        9994 non-null   object        
 2   Order Date      9994 non-null   datetime64[ns]
 3   Ship Date       9994 non-null   datetime64[ns]
 4   Ship Mode       9994 non-null   object        
 5   Customer ID     9994 non-null   object        
 6   Customer Name   9994 non-null   object        
 7   Segment         9994 non-null   object        
 8   Country/Region  9994 non-null   object        
 9   City            9994 non-null   object        
 10  State           9994 non-null   object        
 11  Postal Code     9983 non-null   float64       
 12  Region          9994 non-null   object        
 13  Product ID      9994 non-null   object        
 14  Category        9994 non-null   object        
```

```
# TODO 02 - is there any missing values?, if there is, which colunm? how many nan values?

## ANS: Yes , In Column "Postal Code" Total Missing Value = 11

df.isna().sum()
```

```
Row ID            0
Order ID          0
Order Date        0
Ship Date         0
Ship Mode         0
Customer ID       0
Customer Name     0
Segment           0
Country/Region    0
City              0
State             0
Postal Code      11
Region            0
Product ID        0
Category          0
Sub-Category      0
Product Name      0
Sales             0
Quantity          0
Discount          0
Profit            0
dtype: int64
```

```
# TODO 03 - your friend ask for `California` data, filter it and export csv for him

## ANS: File CSV Name "TODO-03"

df_for_03 = df [df['State'] == "California"]

df_for_03.to_csv('TODO-03.csv')
```

```
# TODO 04 - your friend ask for all order data in `California` and `Texas` in 2017 (look at Order Date), send him csv file

## ANS: File CSV Name "TODO-04"

df_for_04 = df.query( "State == ['California','Texas']" )

df_for_04 = df_for_04[df_for_04 ["Order Date"].dt.year == 2017]

df_for_04.to_csv('TODO-04.csv')
```

```
# TODO 05 - how much total sales, average sales, and standard deviation of sales your company make in 2017

## ANS: Total Sales : 484,247 , Ave Sales : 243 , SD Sales : 753

import numpy as np

df_for_05 = df[df ["Order Date"].dt.year == 2017]

print(np.sum(df_for_05["Sales"]))
print(np.mean(df_for_05["Sales"]))
print(np.std(df_for_05["Sales"]))
```

```
484247.4981
242.97415860511794
753.8641580700248
```

```
# TODO 06 - which Segment has the highest profit in 2018

## ANS: Segment Consumer as 28460

df_for_06 = df[df ["Order Date"].dt.year == 2018]

df_for_06.groupby('Segment')['Profit'].agg(['sum'])
```

	sum
Segment	
Consumer	28460.1665
Corporate	20688.3248
Home Office	12470.1124

```
# TODO 07 - which top 5 States have the least total sales between 15 April 2019 - 31 December 2019

## ANS: (1) New Hampshire / (2) New Mexico / (3) District of Columbia / (4) Louisiana / (5) South Carolina

df_for_07 = df[ (df ["Order Date"] >= '2019-04-15') & (df ["Order Date"] <= '2019-12-31')]

result_07 = df_for_07.groupby('State')['Sales'].agg(['sum'])

result_07.sort_values('sum')
```

	sum
State	
New Hampshire	49.0500
New Mexico	64.0800
District of Columbia	117.0700
Louisiana	249.8000
South Carolina	502.4800
Maine	547.3300
Kansas	691.0600
Iowa	959.3100
Idaho	1148.8060
Delaware	1462.9500
Minnesota	1463.9400
Maryland	1541.0120
Wyoming	1603.1360
Utah	1822.4100
Tennessee	1889.0060
Arkansas	2008.3500
Nebraska	3081.4200
Massachusetts	3489.7340
Missouri	3523.9500
Connecticut	3605.6000
Kentucky	3912.6100
Mississippi	4669.3500
Arizona	4713.9330
Oregon	4914.9600
Oklahoma	5047.6800
Alabama	7651.3300
Georgia	7872.9700
Nevada	9022.6920
Wisconsin	9633.9200
Colorado	10434.4630
North Carolina	11377.0840
Florida	11399.9015
Rhode Island	13085.6000
Illinois	16060.2170
Virginia	16648.3400
New Jersey	17103.3860
Washington	18632.9020
Ohio	23290.4330
Indiana	24844.9700
Michigan	26675.8110
Pennsylvania	28207.2940
Texas	31114.3390
New York	56873.9340
California	105632.9565

```
# TODO 08 - what is the proportion of total sales (%) in West + Central in 2019 e.g. 25%

## ANS: 25% = 17 / 50% = 54 / 75% = 209

df_for_08 = df[df ["Order Date"].dt.year == 2019]

df_for_08 = df.query( "Region == ['West','Central']" )

df_for_08.describe()
```

	Row ID	Postal Code	Sales	Quantity	Discount	Profit
count	5526.000000	5526.000000	5526.000000	5526.000000	5526.000000	5526.000000
mean	5005.030945	80623.579805	221.986557	3.808541	0.164412	26.805069
std	2880.292448	15491.925632	572.687944	2.219430	0.215159	230.965703
min	3.000000	46060.000000	0.444000	1.000000	0.000000	-3701.892800
25%	2528.250000	75023.000000	17.042500	2.000000	0.000000	1.799625
50%	5000.500000	85345.000000	54.264000	3.000000	0.200000	8.715650
75%	7485.750000	93309.000000	209.984250	5.000000	0.200000	28.684875
max	9994.000000	99301.000000	17499.950000	14.000000	0.800000	8399.976000

```
# TODO 09 - find top 10 popular products in terms of number of orders vs. total sales during 2019-2020

## ANS: As below result

df_for_09 = df[ (df ["Order Date"] >= '2019-01-01') & (df ["Order Date"] <= '2020-12-31')]

result_09 = df_for_09.groupby('Product Name')['Sales'].agg(['sum'])

result_09 = result_09.sort_values('sum', ascending=False)

print(result_09.head(10))
```

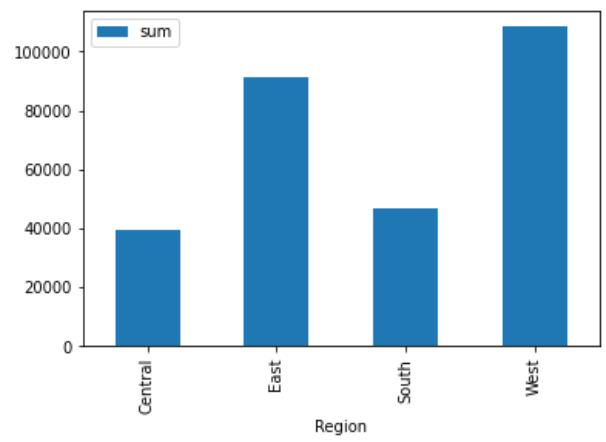
```
                                                    sum
Product Name
Canon imageCLASS 2200 Advanced Copier             61599.824
Hewlett Packard LaserJet 3310 Copier              16079.732
3D Systems Cube Printer, 2nd Generation, Magenta  14299.890
GBC Ibimaster 500 Manual ProClick Binding System  13621.542
GBC DocuBind TL300 Electric Binding System        12737.258
GBC DocuBind P400 Electric Binding System         12521.108
Samsung Galaxy Mega 6.3                           12263.708
HON 5400 Series Task Chairs for Big and Tall      11846.562
Martin Yale Chadless Opener Electric Letter Opener 11825.902
Global Troy Executive Leather Low-Back Tilter     10169.894
```

```
# TODO 10 - plot at least 2 plots, any plot you think interesting :)
```

```
# Plot 1
df.groupby('Region')['Profit'].agg(['sum']).plot( kind = 'bar')
```

```
<Axes: xlabel='Region'>
```

Download

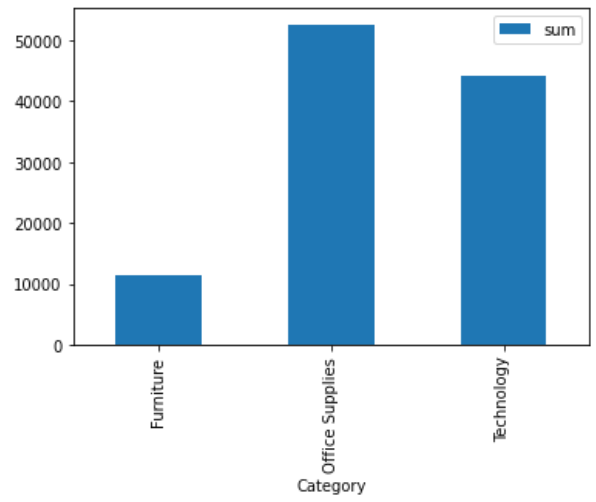


```
# Plot 2

df.query( "Region == ['West']" ).groupby('Category')['Profit'].agg(['sum']).plot(kind='bar')
```

```
<Axes: xlabel='Category'>
```

Download




```
# TODO Bonus - use np.where() to create new column in dataframe to help you answer your own questions

df['Check_Profit'] = np.where(df['Profit'] > 0, "OK", "NG")
```

```
df
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City	...	Region	Product ID	Category	Sub-Category
0	1	CA-2019-152156	2019-11-08	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	South	FUR-BO-10001798	Furniture	Bookcases
1	2	CA-2019-152156	2019-11-08	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	South	FUR-CH-10000454	Furniture	Chairs
2	3	CA-2019-138688	2019-06-12	2019-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	West	OFF-LA-10000240	Office Supplies	Labels
3	4	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	South	FUR-TA-10000577	Furniture	Tables
4	5	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	South	OFF-ST-10000760	Office Supplies	Storage
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9989	9990	CA-2017-110422	2017-01-21	2017-01-23	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Miami	...	South	FUR-FU-10001889	Furniture	Furnishing
9990	9991	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	West	FUR-FU-10000747	Furniture	Furnishing
9991	9992	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	West	TEC-PH-10003645	Technology	Phones
9992	9993	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	West	OFF-PA-10004041	Office Supplies	Paper
9993	9994	CA-2020-119914	2020-05-04	2020-05-09	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westminster	...	West	OFF-AP-10002684	Office Supplies	Appliances

9994 rows × 22 columns